Data in Brief 61 (2025) 111837

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

# Order picking dataset from a warehouse of a footwear manufacturing company

Rodrigo Furlan de Assis, William de Paula Ferreira*, Mustapha Ouhimmou

*Department of Systems Engineering, École de Technologie Supérieure, 1100 Notre Dame Street, West Montreal, QC H3C 1K3, Canada*

A R T I C L E I N F O

*Article history:*
Received 7 January 2025
Revised 30 April 2025
Accepted 23 June 2025
Available online 25 June 2025

Dataset link: Order Picking Dataset from a Warehouse of a Footwear Manufacturing Company (Original data)

*Keywords:*
Warehouse management
Random storage
Mixed shelving
Picker routing

A B S T R A C T

This dataset originates from a real-world footwear manufacturing warehouse and provides a comprehensive foundation for benchmarking research in warehouse order-picking operations. Data was collected via SQL queries on the company's Warehouse Management System (WMS), resulting in diverse formats such as CSV files, CAD layouts, and Python scripts. The dataset includes geometric representations of the warehouse layout, with Cartesian-mapped storage locations, aisles, and central depots, detailed product classifications, storage positions, picking wave information, and routing paths. It supports evaluating various storage strategies, including Random, Class-Based, Dedicated, and Hybrid configurations, enabling the analysis of their impact on order-picking efficiency. Temporal data captures operational trends, including timestamps and operator-specific performance, offering insights into workflow efficiency and workload balancing. Anonymization and randomization techniques were applied while retaining realistic operational patterns to preserve confidentiality. This dataset is highly versatile and suitable for developing optimization algorithms for picker routing, order batching, wave generation, and intralogistics, as well as for advancing automation and robotics research through navigation-specific data for autonomous guided vehicles (AGVs) and robotic systems. This dataset significantly

* Corresponding author.
*E-mail address:* william.ferreira@etsmtl.ca (W. de Paula Ferreira).

https://doi.org/10.1016/j.dib.2025.111837

contributes to warehouse logistics research and operational optimization by supporting a wide range of applications.



Specifications Table

| | |
|---|---|
| Subject | Management Science and Operations Research |
| Specific subject area | Warehouse Management and Intralogistics |
| Type of data | Type(s) of Data: Table, Figure, Image, CAD Layout, Python scripts (.py), CSV Files.<br>Data Formats: Raw, Processed, Filtered, Analyzed. |
| Data collection | Extracted from the company's Warehouse Management System (WMS) using SQL queries. |
| Method of data processing | Data were processed using SQL for extraction, Python for cleaning, statistical analysis, and wave generation, and AutoCAD for layout representation. |
| Data source location | Country: Canada<br>City: Sherbrooke, Province of Quebec. |
| Data accessibility | Repository name: Order Picking Dataset from a Warehouse of a Footwear Manufacturing Company<br>Direct URL to data: https://data.mendeley.com/datasets/pf2w725pw3/1 |
| Related research article | de Assis, R.; de Paula Ferreira, W.; Faria, A.; Santa-Eulalia, L. A.; Ouhimmou, M.; Gharbi, A. (2024). Optimising Warehouse Order Picking: Real Case Application in the Footwear Manufacturing Industry. IEEE Access, 12, 170868–170888. doi:10.1109/ACCESS.2024.3497592. |

## 1. Value of the Data

- Real-World Industrial Relevance: The dataset is based on a footwear manufacturing company, which allows testing and validating algorithms under realistic conditions. Unlike many existing datasets that rely on hypothetical layouts or randomly generated data, this dataset captures real warehouse operations, including precise storage locations, product inventory levels, and picking orders, which enables research into warehouse optimisation and intralogistics.
- Geometric Warehouse Representation: Includes a CAD-based geometric model of the warehouse, providing precise visualization of aisles, storage locations, and the central depot. Storage locations are defined with Cartesian coordinates and formatted for seamless export to diverse platforms, enabling simulations, algorithm development, and advanced routing strategy modelling.
- Granular Picking and Routing Data: Contains CSV files extracted from the WMS, detailing product locations, stock levels, picking sequences, and routing distances. These data enable the evaluation of picking strategies, optimisation of routes, and testing of algorithms, while their granularity facilitates benchmarking and validation against real-world operational scenarios.
- Product and Inventory Details: Offers comprehensive product classifications, stock quantities, and associated storage locations. This information supports the study of inventory management methods, including storage location optimisation, analysis of stocking patterns, and retrieval efficiency. These elements are critical for developing strategies that enhance picking performance.
- Operational Traceability of Picking Waves: Provides structured data linking each picking wave to the corresponding items, storage locations, and assigned operators. While the dataset does not include timestamps or continuous tracking, the recorded pick sequence supports the re-

construction of picking paths and enables the analysis of routing patterns and workload distribution.
- Support for Automation and Robotics Research: Includes navigation-specific data, such as predefined support points and optimised paths, suitable for developing algorithms for autonomous guided vehicles (AGVs) and robotic systems. These data are integral to designing systems to enhance movement efficiency, minimise travel distances, and reduce collision risks in warehouse operations.
- Versatile Formats and Broad Applicability: Delivered in CSV, CAD, and PDF formats, ensuring compatibility with analytical and operational tools. The structure is tailored to support studies in order-picking optimization, warehouse layout design, and intralogistics planning, aligning with industrial engineering, operations research, and robotics domains.

## 2. Background

This study presents a dataset from a semi-automated footwear manufacturing warehouse designed to evaluate optimisation approaches for picking operations. The warehouse employs a random storage system, allowing multiple products per location to maximise space utilisation while posing challenges for route planning. It features multi-level racking: levels 1 and 2 for manual picking and levels 3 and 4 as replenishment zones accessed by forklifts. The layout includes parallel and cross aisles, with a single entry/exit point, and operators use trolleys to navigate demand-driven picking waves. The dataset comprises geometric representations of the warehouse, Cartesian-mapped product locations and temporal data on order preparation. It enables the evaluation of optimisation algorithms and the validation of models. It addresses constraints like variable product placement and item overlap to support practical solutions for travel distance reduction, storage optimisation, and intralogistics improvement. Recent reviews have emphasised the scarcity of real-world datasets that capture detailed warehouse operations, particularly for storage location assignment and order picking problems [1]. By providing structured and realistic data extracted from an operational environment, this dataset offers valuable support for developing and validating optimisation and machine learning models in warehouse management research.

## 3. Data Description

This section provides a detailed description of the datasets used in this study, which were derived from a semi-automated warehouse in the footwear manufacturing industry. The data include formats like CAD files for the geometric layout, CSV files containing product, location, and picking wave information, and temporal data capturing order preparation and operator performance. The dataset was adapted using anonymisation and randomisation techniques to protect the company's confidential information while maintaining its representativeness for research purposes. These adjustments ensure that operational and structural details remain realistic without compromising proprietary data.

### *3.1. Warehouse Configuration*

The warehouse layout, provided in CAD format, offers a detailed geometric representation for understanding and optimising order-picking routes. Concerning picker routing, the main objective is to minimise the total distance travelled due to its high proportion in total picking time [2–4]. The facility operates under a random storage policy to maximise space utilisation, allowing up to 18 different products per location. A product classification system tracks item references, sizes, and attributes, enabling operators to locate products efficiently despite the randomness of the storage arrangement.

**Table 1**
Overview of the warehouse layout.

| Parameter | Value | Units |
|---|---|---|
| Number of levels | 4 | Unit |
| Number of racks | 25 | Unit |
| Total vertical Aisles | 3 | Unit |
| Total parallel Aisles | 17 | Unit |
| Capacity of a storage location | 18 | Unit |
| Total storage location | 2293 | Unit |
| Products range per pick list | 15 to 27 products | Unit |

The warehouse consists of multiple blocks organised into aisles and shelves, allowing efficient product access during picking, as shown in Table 1.

The warehouse has shelving units structured on four vertical levels, each with a base area of 3.66 m long and 1.52 m wide. The two upper levels, used for stock replenishment, have a vertical height of 1.79 m each and are designed to accommodate large boxes, with each position supporting up to 1,293 kg. The two lower levels, designated for picking operations, have a height of 1.03 m each and are subdivided horizontally into three compartments, resulting in three picking positions per shelving unit on the first and second levels. Each compartment is sized to hold small boxes and supports a load of up to 612 kg per level.

Products are stored in two packaging formats aligned with the shelf configuration: large boxes measuring 0.787 m x 0.470 m x 0.416 m are placed in the replenishment zones, while small boxes measuring 0.381 m x 0.381 m x 0.152 m are stored in the picking compartments. Each large box can contain up to six small boxes. Depending on the boot model, product weights vary: lightweight items (short shaft, composite) average 1.2 kg, medium-weight models (standard leather with regular toe cap) weigh around 1.8 kg, and heavier products (high shaft, steel toe, insulated) can reach 2.6 kg. These variations directly affect storage allocation and order-picking strategies.

### *3.2. General data*

Furthermore, additional data can be accessed in the repository mentioned in the table (Data Specifications), which consists of four main CSV files:

- Product.csv
- Storage_Location.csv
- Random_Storage.csv
- Class_Based_Storage.csv
- Dedicated_Storage.csv
- Hybrid_Storage.csv
- Support_Points_Navigation.csv
- Picking_Wave.csv

The Product.csv file contains detailed information about each product stored in the warehouse. The products are related to the footwear manufacturing sector, specialising in a wide range of high-quality boots designed for the construction, mining and manufacturing sectors. Each product (box of two footwear) is identified by a unique reference code ("Reference") corresponding to a specific item, ensuring accurate tracking and stock management.

The Storage_Location.csv file describes the storage points within the warehouse, specifying the exact location of each product. The Location column identifies the aisle and position of each storage point, such as "A-14-11" or "A-18-22", indicating the corresponding block, column, and shelf. The Points column provides the three-dimensional coordinates of each location (X, Y, Z), allowing for a precise visualisation of the positions. Segmented into discrete storage positions, the shelves are labelled using a system with a unique alphanumeric code that indicates each storage position. For example, a typical representation would be as follows: J-10-21, where:

- J indicates aisle J.
- 10 indicates that it is in the 10th column of aisle J.
- 2 indicates it is on level 2 (second floor).
- 1 indicates that this is the first location in this column and on this floor. (There can be several locations in the same column and on the same floor, as with almost all locations on floors 1 and 2).

To evaluate the impact of different storage strategies on order-picking efficiency, four CSV files are provided, each simulating a specific storage policy. Each file follows the same structure and represents a fully populated warehouse inventory designed for optimisation studies in warehouse management. The files contain 18 columns, each corresponding to a specific storage location. Each cell is filled with a product identifier ("Reference") and a size attribute ("Size"), formatted as "Reference;Size". The size values range from 4 to 16, simulating variations in product dimensions found in real-world warehouse operations. The content of each file is generated based on the rules of its respective storage policy:

- Random_Storage.csv: Products are randomly distributed across all available storage positions.
- Class_Based_Storage.csv: Products are categorised by demand class (A, B, C), with high-demand items placed closer to the entry/exit (I/O) point.
- Dedicated_Storage.csv: Products are ranked by demand and assigned to fixed positions optimised for travel distance to the I/O point.
- Hybrid_Storage.csv: Combines Dedicated and Random storage, allocating high-demand products to 20 % of the positions near the I/O point, while the remaining positions are randomly filled within demand classes.

Generating the different storage strategies followed a structured methodology based on historical product demand. An ABC analysis was applied for the Class-Based Storage configuration, ranking products according to picking frequency. Class A products represented the top 20 % of cumulative demand, Class B the next 30 %, and Class C the remaining 50 %. Products were then located to storage locations based on proximity to the I/O point. Class A products were positioned closest, Class B in intermediate zones, and Class C farther away, simulating a demand-based slotting strategy commonly used in warehouse management.

For the Dedicated Storage strategy, products were individually ranked by demand frequency, and each was assigned to a specific, fixed location to minimise the expected travel distance from the I/O point. The Hybrid Storage configuration combined these approaches by reserving 20 % of the nearest storage locations for high-demand products in a dedicated manner. At the same time, the remaining 80 % of the warehouse was populated through a random location within each demand class.

The Support_Points_Navigation.csv provides a detailed map of support points to navigate the warehouse efficiently. Each support point is associated with specific coordinates to follow optimised paths during picking operations. The dataset includes the spatial coordinates of each point and a label that classifies the points according to their position within the warehouse layout. The nomenclature for the support points is as follows:

- Left Corridor (LC): Left-hand side of the warehouse, primarily used for guiding movement through the left-most aisles.
- Central Corridor (CC): Central part of the warehouse, used for guiding movement through the middle aisles.
- Right Corridor (RC): Right-hand side of the warehouse, assisting in navigating through the right-most aisles.

Each support point is associated with its corresponding coordinate, represented in three dimensions (X, Y, Z). This geometrical information is crucial for developing and optimising naviga-

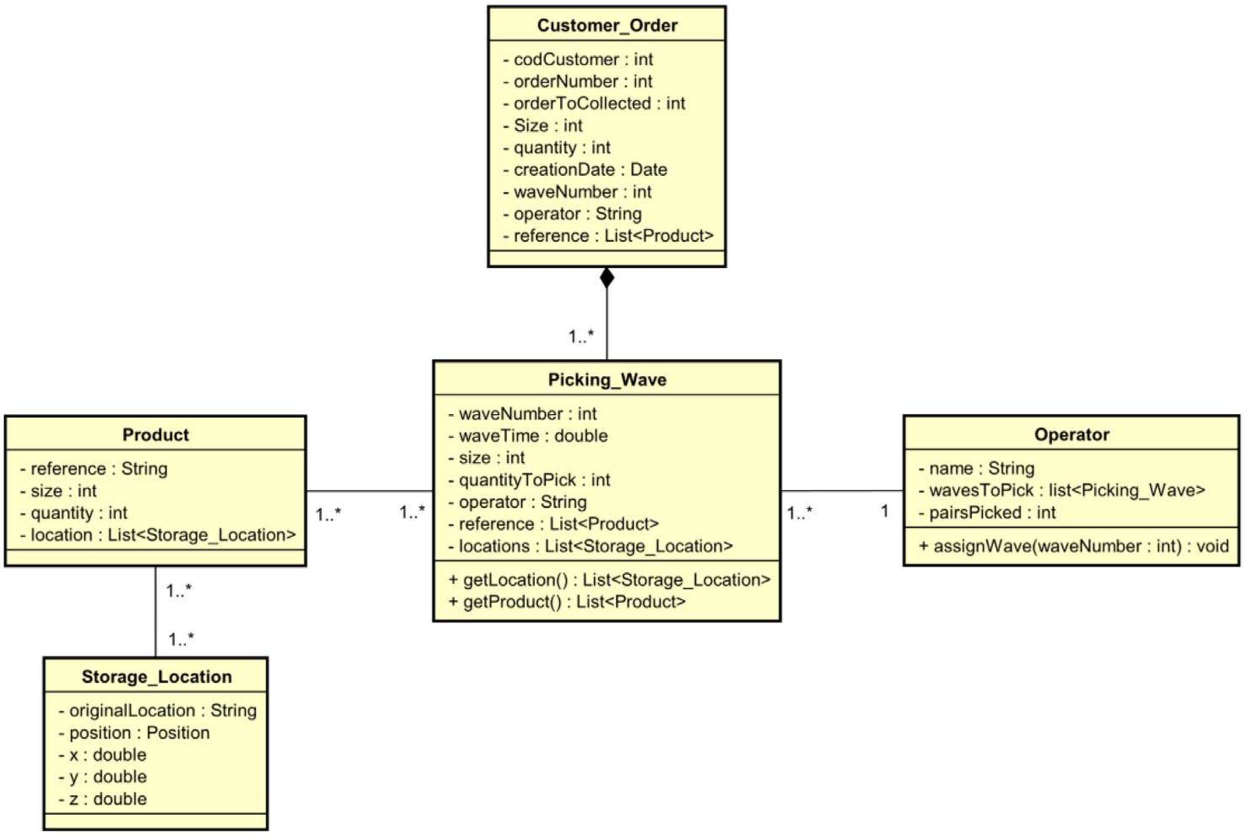


**Fig. 1.** UML class diagram.

tion algorithms. Additionally, this dataset is valuable for analysing movement efficiency within the warehouse, ensuring that routes are streamlined and optimised for the shortest possible travel distances during picking operations.

The Picking_Wave.csv file offers detailed information on the warehouse's operational aspects of picking waves. It includes the duration of each wave, the number of products picked, and their respective locations. This data is a foundation for simulating order-picking scenarios and testing algorithms across different configurations and demand patterns. Additionally, it specifies the products to be collected, their storage locations, and the assigned operators, providing a comprehensive view of the warehouse's picking operations.

The dataset provides detailed information for each picking wave, including product reference, size, quantity to pick, storage location, and assigned operator. Picking waves are constrained to a maximum of 27 items, corresponding to the physical capacity of the trolleys used in the picking operation. Each wave may involve either single-line orders (comprising one product reference) or multi-line orders (involving multiple product references), supporting structured analyses of order complexity, batching policies, and picker workload balancing.

Fig. 1 illustrates a UML class diagram that models the entities and relationships essential for managing orders, products, and picking waves in the warehouse environment.

The UML diagram captures the main entities and relationships in the warehouse picking system. The central class, **Picking_Wave**, defines the picking waves linked to **Customer_Order** through a one-to-many relationship, indicating that a wave may handle multiple customer orders. Each order now includes attributes such as **orderNumber, codCustomer, orderToCollect**, and quantity, and refers to a list of Product instances. Products are associated with their respective **Storage_Location**, which now includes attributes like **originalLocation**, Cartesian coordinates (X, Y, Z), and position. Each storage location can hold multiple products, reflecting the flexibility of the storage system.

The **Picking_Wave** class now tracks details such as size and **quantityToPick**, supporting the wave's role in consolidating products for orders. The Operator class remains connected to **Picking_Wave** through a one-to-many relationship, representing the allocation of waves to operators.

Operators are monitored by attributes such as **wavesToPick** and **pairsPicked**, which reflect their performance metrics.

## 4. Experimental Design, Materials and Methods

### *4.1. Design*

The dataset was developed to investigate route optimisation for picking operations in a warehouse environment, where distance travelled is the primary analysis factor. Each collection and delivery point in the warehouse was mapped on a Cartesian plane, with coordinates assigned to represent the exact location of each point in space. To configure the dataset, the following elements were defined:

- Picking and Delivery Points (Nodes): Each picking and delivery point was treated as a node within a Cartesian coordinate system, where coordinates (X, Y) reflect its location within the warehouse. These nodes were distributed to mimic different areas and sections of the actual storage space.
- Wave Generation: The picking waves were generated based on real data provided by the company, simulating different picking orders and reflecting actual demand patterns. This wave-based structuring enables the evaluation of travel distance for picking scenarios.

The dataset analyses the impact of distance across picking configurations, with Cartesian coordinates assigned to nodes and wave-based ordering reflecting warehouse dynamics. Data from CSV files, such as support points and storage locations, was used to configure the layout, adding navigation corridors to represent aisles. The process, implemented in Python, includes code for data processing, layout visualisation, and export. It is available in the repository for reproducibility and further research.

In addition to the elements described the Cartesian coordinates of each collection and delivery point strictly follow the measurements found in the CAD file, guaranteeing geometric accuracy. These points were defined using Algorithm 1 to generate and save the coordinates in a CSV spreadsheet.

Algorithm 1 for generating and saving coordinates begins by loading a CSV file and defining a function to create coordinates based on initial values and a specified increment for the x-axis. Using parameters for starting coordinates and the number of points, the function generates a sequence of (X, Y, Z) coordinates, adjusting each subsequent point by decrementing the X-value. The generated coordinates are then added as a new column to the original CSV file, which is saved to a new path. This approach ensures the geometrical positions reflect the CAD layout, facilitating accurate visualisation and distance calculations in Python.

**Algorithm 1** Generate and save coordinates in CSV.

**Load** the CSV file at path *file_path* into DataFrame **df**.
**Define Function** *generate_coordinates* with parameters:
1. **X_start, Y_start, Z_start**: initial coordinates
2. **num_points**: number of points to generate
3. **X_increment**: decrement for each X-coordinate.

**Initialise** empty list *coordinates*.
**for** each point from 1 to *num_points* do:
1. Append coordinate (*current_X, current_Y, current_Z*) to coordinates.
2. Decrease *current_X* by *X_increment*.
3. **end for**

**Set Initial Values**: *X_start, Y_start, Z_start, X_increment*.
**Generate Coordinates** using *generate_coordinates* for the number of rows in **df**.
**Add Column** "Points" in **df** with *generated coordinates*.
**Save df** to new CSV file at path *updated_file_path*.

### 4.2. Materials

The materials used to develop this dataset include resources to ensure accurate geometric representation and facilitate the analysis of picking operations within the warehouse.

- CAD File: The CAD layout of the warehouse provided precise geometric data for defining each storage location and picking points in Cartesian coordinates. This file included measurements of aisles, shelves, and storage zones, serving as the foundation for the geometric representation of the warehouse.
- Data from the WMS: Operational data, such as product locations, stock levels, and picking wave details, were extracted directly from the company's WMS through structured SQL queries. The extracted data were cleaned and formatted into CSV files, ensuring consistency with real-world operational parameters and accurately reflecting the warehouse's storage and order-picking configurations.
- Wave Generation for Picking Orders: The picking waves were generated based on actual order data provided by the company, representing realistic demand patterns. This process involved structuring the waves chronologically, reflecting the operational workflow without applying optimisation techniques. Each wave included detailed product lists, their respective storage locations, and timestamps, enabling the simulation of different order-picking scenarios under realistic operational conditions.
- Python Environment: Data processing and warehouse layout visualisation were performed using Python, leveraging its extensive libraries for computational and graphical tasks:
  - Pandas: Used for loading, cleaning, and manipulating the data extracted from the WMS, enabling structured handling of tabular information such as product classifications, locations, and wave details.
  - Matplotlib: Employed to visualise the warehouse layout, including the representation of storage racks, aisles, and picking paths. Each shelf was depicted as a rectangle at its Cartesian coordinates, matching the dimensions derived from the CAD file.
  - CSV Library: Utilised for efficiently reading and writing tabular data in CSV format, ensuring compatibility between raw data and processed datasets.
- Algorithm for Generating Coordinates: A custom algorithm was developed to assign Cartesian coordinates to each storage location and pick points based on the CAD layout. The algorithm systematically computed the coordinates by iterating over predefined grid patterns, applying incremental adjustments to the X, Y, and Z axes.
- PDF Export Functionality: The final warehouse layout was exported as a high-resolution PDF file after being processed and visualised in Python.

### 4.3. Methods

The dataset was constructed using steps to ensure spatial representation, accurate wave generation, and practical visualisation, all reflecting the warehouse's real operational environment. This methodological approach enables a dataset by combining CAD-based coordinate generation, real-data-driven wave sequencing, and precise layout visualisation. This dataset supports comprehensive testing of distance-minimisation algorithms within a structured and realistic warehouse framework, facilitating operational optimisation and algorithmic development.

#### 4.3.1. Coordinate Assignment and Geometric Mapping

The CAD layout of the warehouse provided precise geometric dimensions for each storage location and picking station, serving as the foundation for assigning Cartesian coordinates. Using Algorithm 1, these positions were systematically mapped to ensure accuracy and alignment with the physical structure of the warehouse.

The algorithm initialises starting coordinates (*X_start, Y_start, Z_start*) based on the CAD layout and iteratively updates the coordinates for each node. The X value is decremented by a pre-

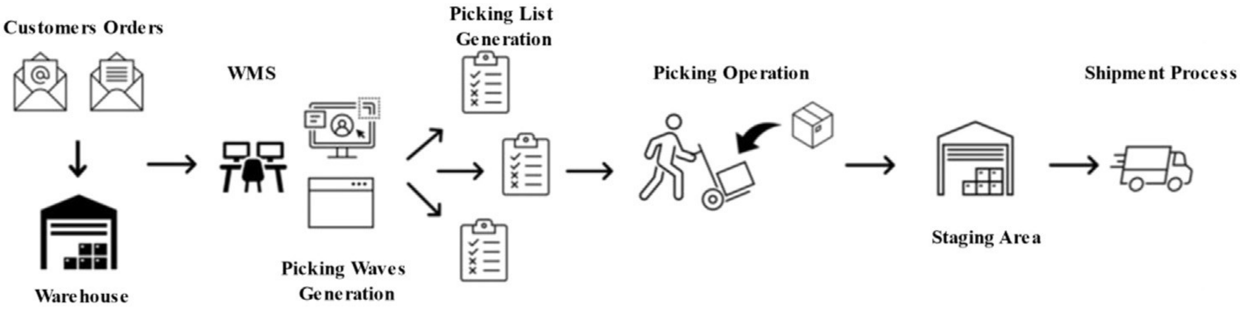


**Fig. 2.** Operating deliveries flow.

defined *X_increment*, while *Y* and *Z* remain constant, replicating the warehouse's arrangement of aisles and shelves. The resulting coordinates are stored in a structured CSV file, facilitating integration with wave generation and layout visualisation processes.

#### *4.3.2. Wave Generation for Order Sequencing*

The generation of picking waves in the warehouse is directly linked to processing customer orders received daily. The overall operational flow is illustrated in Fig. 2.

The orders are transmitted to the Warehouse and consolidated into a central database. The process begins with the WMS, which generates picking waves by organising tasks based on the sequence of order reception and operational schedules without applying specific grouping or optimisation techniques. These waves are then converted into picking lists detailing the products to be retrieved and their respective storage locations. The picking operation follows these lists, ensuring products are collected and moved to the Staging Area for final consolidation before shipment.

#### *4.3.3. Warehouse Layout Visualisation*

The warehouse layout was visualised using Python, with Pandas for data manipulation and Matplotlib for spatial rendering. Key operational features include:

- Position Rendering}: Each storage location was represented as a rectangle plotted at its assigned Cartesian coordinates, with the position and size of each rectangle scaled to mirror the dimensions in the CAD layout.
- Navigation Corridors: Using dashed lines, corridors were drawn to represent navigational pathways between storage rows, emulating the physical aisles and facilitating understanding of operator movement within the warehouse layout.
- Adjustment for Visual Clarity: Adjustments were made to ensure that elements did not overlap. Labs were added where necessary to demarcate each section, enhancing the layout's readability for analysis.

This visualisation allows routing algorithms to be tested in a realistic warehouse environment, providing immediate feedback on the positioning of nodes and the path structure. Fig. 3 illustrates the layout of the first floor, organised into three main blocks (A, B and C) with double-sided shelves. The design features parallel sorting aisles connected by cross aisles at both ends and a central cross aisle that divides the space into distinct sorting sub-areas. The diagram shows that the intersection points simplify access and navigation through the aisles. Each storage location is designated for specific product picking operations, while the Warehouse (I/O) is the central hub for operational workflows.

#### *4.3.4. PDF Export of Geometric Layout*

After visualisation, the warehouse layout was exported as a high-resolution PDF file. This export provides an accurate and static representation of the geometric configuration, serving as a reliable reference for documentation and validation purposes. The PDF format ensures compatibility across various platforms, making it accessible to technical and non-technical stakeholders.

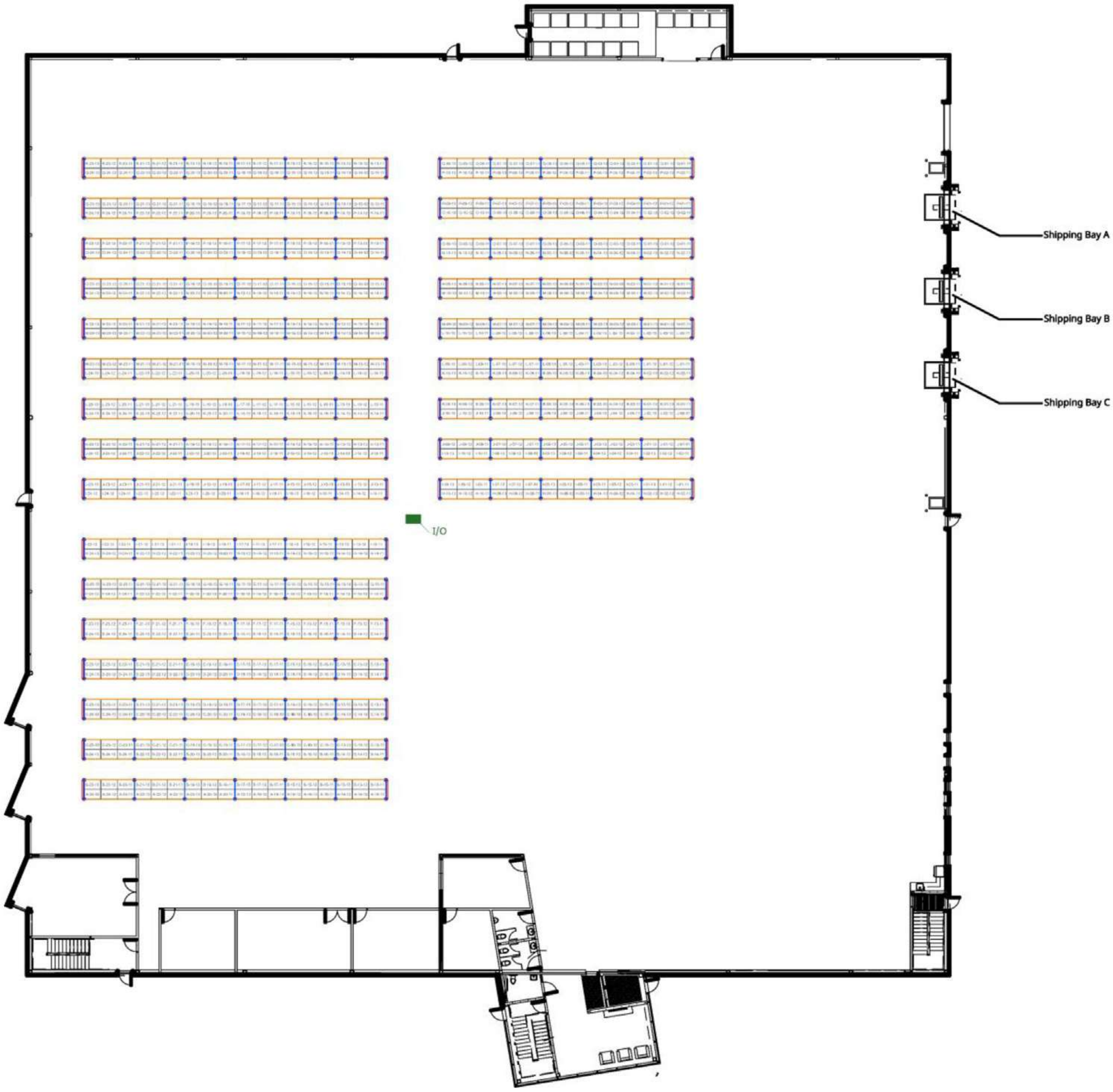


**Fig. 3.** Schematic representation of the warehouse layout, highlighting the storage locations and picking positions of the first level.

## 5. Potential Research Application

The dataset offers a foundation for warehouse management, operations research, and automation research applications. One important line of study involves the efficiency comparison of different picking strategies, such as single-order, batch, and zone picking, under realistic warehouse conditions. Researchers can simulate operational scenarios and assess travel distance, order cycle time, and picker workload performance by leveraging detailed picking wave information and geometric layout data. Furthermore, the dataset enables the evaluation of different storage assignment policies — including Random, Class-Based, Dedicated, and Hybrid strategies — providing empirical evidence of their impact on operational efficiency.

In addition to storage and picking strategy analysis, the dataset supports developing and validating picking path optimisation algorithms. The detailed mapping of storage locations and support points allows researchers to formulate and test optimisation models, such as adaptations of the Travelling Salesman Problem (TSP), tailored explicitly to complex warehouse environments. Temporal data associated with order preparation and operator performance further enables studies focused on workload balancing and task allocation strategies, promoting the design of solutions that enhance efficiency and fairness in warehouse operations.

Finally, the dataset offers opportunities for research in warehouse automation and robotics. Including navigation-specific support points and spatially accurate representations of aisles and storage, zones are suitable for developing navigation algorithms for autonomous guided vehicles (AGVs) and robotic picking systems. This level of detail facilitates testing real-time routing algorithms and autonomous path planning strategies, contributing to advancing intralogistics solutions. The dataset provides an environment for benchmarking, simulation, and algorithm development, supporting research efforts to optimise warehouse operations in human-operated and automated contexts.

## Limitations

Although the dataset provides a detailed and realistic basis for research into picking optimisation in warehouses, some limitations should be acknowledged. The data was collected from a single footwear manufacturing warehouse, which may restrict the generalisability of the results to other sectors or warehouse configurations. Although anonymisation and randomisation techniques were carefully applied to protect confidential operational information, this process may have led to subtle operational losses. However, efforts were made to retain key operational characteristics, including spatial layouts, statistical product storage distributions and separation wave structures. Consequently, the dataset retains high fidelity for research applications related to warehouse logistics, order picking and intralogistics optimisation. Future extensions could involve integrating data from several warehouses in different sectors to broaden the scope and representativeness of the dataset.

Additionally, it should be noted that although the dataset includes detailed storage locations and navigation points, the WMS does not record the continuous spatial trajectories of pickers during operations. However, since the system logs the sequence in which each item is picked within a wave, it is possible to reconstruct the picker's path using the ordered list of storage locations and to estimate picker speed by calculating the total distance travelled. Nevertheless, the dataset does not provide pre-calculated performance indicators such as picking times or error rates. This information enables route simulation and supports the development of optimisation algorithms, although it does not represent exact real-time movement trajectories.

## Ethics Statement

The authors confirm that they have adhered to the ethical guidelines required for publication in Data in Brief. Additionally, this work does not involve research with human subjects, animal experiments, or the use of data obtained from social media platforms.

## Data Availability

Order Picking Dataset from a Warehouse of a Footwear Manufacturing Company (Original data) (Mendeley Data).

## CRediT Author Statement

**Rodrigo Furlan de Assis:** Conceptualization, Methodology, Investigation, Data curation, Formal analysis, Visualization, Writing – original draft, Writing – review & editing; **William de**

**Paula Ferreira:** Conceptualization, Methodology, Investigation, Data curation, Formal analysis, Visualization, Writing – review & editing, Resources, Validation, Funding acquisition, Supervision; **Mustapha Ouhimmou:** Funding acquisition, Validation, Writing – review & editing.

## Acknowledgments

This research was supported by Mitacs through the Mitacs Accelerate program in Canada and by the Natural Sciences and Engineering Research Council (NSERC) of Canada [grant number RGPIN-2024-03880].

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] R.F. Assis, A.F. Faria, V. Thomasset-Laperrière, L.A. Santa-Eulalia, M. Ouhimmou, W.P. Ferreira, Machine learning in warehouse management: A survey, Procedia Comput. Sci. 232 (2024) 2790–2799.
[2] T.V. Gils, K. Ramaekers, An Caris, RBMDe Koster, Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review, Eur. J. Oper. Res. 267 (1) (2018) 1–15.
[3] M. Masae, C.H. Glock, E.H. Grosse, Order picker routing in warehouses: A systematic literature review, Int. J. Prod. Econ. 224 (2020) 107564.
[4] R.F. Assis, W.P. Ferreira, A.F. Faria, L.A. Santa-Eulalia, M. Ouhimmou, A. Gharbi, Optimising warehouse order picking: Real case application in the shoe manufacturing industry, IEEe Access 12 (1) (2024) 170868–170888.